STRUKTUR DATA DAN ALGORITMA BEROIENTASI OBJEK PROGRAM BUAH (DATA STRUCTURES AND OBJECT ORIENTED ALGORITHMS FRUIT PROGRAMS)

Sekar Wangi Qurrota Aini[1],Nurun Nafiah[2],Yuniana Cahyaningrum[3]

IKIP PGRI BOJONEGORO

Email : sekaraini270904@gmail.com, nurunnafiah14@gmail.com,yuniana@ikippgribojonegoro.ac.id

Abstract*: The data structure is a way to save and manage data in a computer so that it can be accessed and manipulated efficiently. Examples of data structures include arrays, Linked List, Stack, Queue, Tree, and Graph. Each data structure has the advantages and disadvantages of each depending on the type of operation that will be carried out in the data. Meanwhile, the algorithm is a logical instruction sequence applied to data to solve problems or achieve certain goals. Algorithms can be written in various programming languages and can be applied to various types of data structures. Some general algorithms include sorting, searching (searching), traversal, and graph processing (graph processing). Both concepts can be used together to solve various problems in programming. Object-oriented approaches (OPIENTED PROGRAMMING or OOP) can also be applied to organize data structures and algorithms in a program. In OOP, the data structure and algorithm are represented as objects and methods that interact with each other to achieve certain goals. This approach can help increase modularity, flexibility, and readability of the program.*

Keyword : *array, linked list, stack , queue, tree*

Abstrak : Struktur data adalah cara untuk menyimpan dan mengatur data dalam komputer agar dapat diakses dan dimanipulasi dengan efisien. Contoh struktur data termasuk array, linked list, stack, queue, tree, dan graph. Setiap struktur data memiliki kelebihan dan kekurangan masing-masing tergantung pada jenis operasi yang akan dilakukan pada data tersebut. Algoritma adalah urutan instruksi logis yang diterapkan pada data untuk memecahkan masalah atau mencapai tujuan tertentu. Algoritma dapat ditulis dalam berbagai bahasa pemrograman dan dapat diterapkan pada berbagai jenis struktur data. Beberapa algoritma umum meliputi pengurutan (sorting), pencarian (searching), traversal, dan pengolahan grafik (graph processing).Kedua konsep tersebut dapat digunakan bersamaan untuk memecahkan berbagai masalah dalam pemrograman. Pendekatan berorientasi objek (object-oriented programming atau OOP) juga dapat diterapkan untuk mengorganisir struktur data dan algoritma dalam sebuah program. Dalam OOP, struktur data dan algoritma diwakili sebagai objek dan metode yang saling berinteraksi untuk

mencapai tujuan tertentu. Pendekatan ini dapat membantu meningkatkan modularitas, fleksibilitas, dan keterbacaan program.

kata kunci :array, linked list, stack , queue, tree

PENDAHULUAN

Structur data dan algoritma berorientasi objek (OOP) adalah dua konsep yang penting dalam pemrograman. Struktur data adalah cara untuk menyimpan dan mengorganisir data di dalam sebuah program, sedangkan algoritma adalah cara untuk memanipulasi data tersebut agar sesuai dengan kebutuhan. (Georgius Rinaldo , 2014)

Dalam OOP, struktur data dan algoritma dibuat dengan menggunakan konsep objek. Objek adalah representasi dari entitas nyata atau konsep dalam program. Setiap objek memiliki atribut atau properti yang merepresentasikan karakteristik dari objek tersebut, dan metode atau fungsi yang merepresentasikan perilaku objek tersebut (Georgius Rinaldo , 2014).

Salah satu contoh struktur data dalam OOP adalah array. Array adalah tipe data yang digunakan untuk menyimpan sekumpulan nilai yang memiliki tipe data yang sama. Array dapat diakses dengan menggunakan indeks, yang mengacu pada posisi nilai dalam array tersebut. (Georgius Rinaldo , 2014).

Sedangkan contoh algoritma dalam OOP adalah pengurutan data. Algoritma pengurutan adalah cara untuk menyusun sekumpulan data dalam urutan tertentu, seperti dari yang terkecil hingga yang terbesar atau sebaliknya. Dalam OOP, pengurutan dapat dilakukan dengan membuat objek yang merepresentasikan data yang akan diurutkan, dan kemudian menggunakan metode yang sesuai untuk mengurutkan objek tersebut.

OOP memungkinkan programmer untuk membuat struktur data dan algoritma yang lebih kompleks dan mudah diatur. Dengan menggunakan konsep objek, programmer dapat membuat kode yang lebih mudah dimengerti, dipelihara, dan digunakan ulang. Oleh karena itu, OOP menjadi sangat populer dalam pengembangan perangkat lunak modern.

TINJAUAN PUSTAKA

Struktur Data adalah cara mengumpulkan dan mengatur data sedemikian rupa sehingga kita dapat melakukan operasi pada sebuah data dengan cara yang efektif. Struktur Data adalah tentang merender elemen data dalam beberapa hubungan, untuk organisasi dan penyimpanan yang lebih baik. Sebagai contoh, kami memiliki beberapa data yang memiliki, nama pemain “Virat” dan usia 26. Di sini “Virat” adalah tipe data String dan 26 adalah tipe data integer.Kami dapat mengatur data ini sebagai catatan seperti catatan Pemain, yang akan memiliki nama dan usia pemain di dalamnya. Sekarang kita dapat mengumpulkan dan menyimpan catatan pemain dalam file atau database sebagai struktur data. Misalnya: “Dhoni” 30, “Gambhir” 31, “Sehwag” 33Jika Kamu mengetahui konsep pemrograman Berorientasi Objek, maka Class juga melakukan hal yang sama, ia mengumpulkan berbagai jenis data di bawah satu entitas tunggal. Satu-satunya perbedaan adalah, struktur data menyediakan teknik untuk mengakses dan memanipulasi data secara efisien.

Dalam bahasa sederhana, Struktur Data adalah struktur yang diprogram untuk menyimpan data yang dipesan, sehingga berbagai operasi dapat dilakukan dengan mudah. Ini mewakili pengetahuan tentang data yang akan diatur dalam memori. Ini harus dirancang dan diimplementasikan sedemikian rupa sehingga mengurangi kompleksitas dan meningkatkan efisiensi.Segala sesuatu yang dapat menyimpan data dapat disebut struktur data, maka Integer, Float, Boolean, Char dll, semuanya adalah struktur data. Mereka dikenal sebagai Struktur Data Primitif.

A. Tipe Struktur Data :

1. Graph

   Tipe ini cukup sering digunakan dalam berbagai kesempatan. Graph sendiri merupakan struktur data non-linear, yang terdiri dari beberapa node yang saling terhubung. Setiap garis akan saling menghubungkan dua simpul. Contohnya seperti jaringan telepon.

2. Tree

   Tipe tree ini diibaratkan sebagai akar dari sebuah pohon. Tree dimaknai sebagai kumpulan node, dimana masing-masing node tersusun dari value dan juga berbagai referensi. Sederhananya, setiap node berisi beberapa data atau link dari node lainnya.

3. Array

   Array adalah susunan, yang mana tipe ini adalah struktur data yang disusun secara linear dan berdekatan. Umumnya, data yang disusun memiliki jenis yang sama sehingga pengguna data dapat dengan mudah memilah data berdasarkan tipenya.

4. Linked List

   Linked List adalah struktur data yang terdiri dari urutan data linear, yang dihubungkan satu sama lain. Linked list terbagi menjadi tiga jenis, yaitu singly linked list, doubly linked list, dan circular linked list.

5. Queue

   Queue menggunakan sistem FIFO atau First In First Out, dalam artian tipe ini adalah struktur data linear yang cara kerjanya seperti antrian. Data queue tersusun dalam suatu urutan, data yang pertama kali masuk ke dalam urutan merupakan data yang diproses duluan.

6. Stack

   Tipe ini merupakan struktur data linear dan mengikuti urutan tertentu. Urutan yang digunakan adalah Last In First Out (LIFO) atau First In Last Out (FILO). Keduanya memiliki arti yang sama, yaitu data yang terakhir masuk akan menjadi data yang keluar pertama kali.

7. Hash Table

   Hash Table adalah tipe yang digunakan untuk menyimpan data secara asosiatif. Tipe ini menyimpan data dalam

format array. Ini memungkinkan Anda untuk mengakses data dengan cepat, karena cukup menggunakan indeksnya saja.

B. Manfaat Struktur Data

- Membuat proses pemrograman yang mudah. Struktur data dapat digunakan untuk membuat perintah melalui bahasa pemrograman secara mudah.
- Menyimpan dan mengatur data. Manfaat penerapan struktur data yakni kamu jadi lebih mudah menyimpan dan mengatur data secara efisien dan terorganisir.
- Pertukaran data menjadi lebih mudah karena kamu bisa membagi informasi dengan cepat antar aplikasi.
- Meningkatkan kualitas algoritma. Dengan menggunakan struktur data dan menerapkan penyusunan data yang terorganisir, maka dapat berpengaruh pada tingkat efektivitas algoritmanya.
- Mengantisipasi masalah coding. Ketika menggunakan struktur data dan kamu menemukan error, maka error tersebut bisa lebih cepat ditemukan.
- Mengelola sumber daya dan layanan. Sumber daya dan layanan dalam sistem operasi dapat diatur melalui struktur data, struktur data tersebut biasanya memuat manajemen direktori file, alokasi memori, dan masih banyak lagi.
- Pengindeksan. Struktur data dapat digunakan untuk mengindeks objek yang disimpan pada database.
- Searching. Indeks yang terdapat pada struktur data dapat mempercepat proses pencarian data tertentu.
- Skalabilitas. Struktur data digunakan untuk mengalokasikan serta mengelola penyimpanan data di seluruh lokasi penyimpanan. Sehingga sebuah aplikasi big data dapat terjamin performa dan kemampuan upgrade-nya.

Asal kata Algoritma berasal dari nama Abu Ja’far Mohammed Ibn Musa al-Khowarizmi, ilmuan Persia yang menulis kitab al jabr w’al-muqabala (rules of restoration and reduction) sekitar tahun 825 M Definisi Algoritma -Algoritma adalah urutan langkah logis tertentu untuk memecahkan suatu masalah. Yang

ditekankan adalah urutan langkah logis, yang berarti algoritma harus mengikuti suatu urutan tertentu, tidak boleh melompat-lompat. (Dari Microsoft Press Computer and Internet Dictionaary 1997, 1998).

Algoritma adalah barisan langkah-langkah perhitungan dasar yang mengubah masukan (dari beberapa fungsi matematika) menjadi keluaran.

Alat yang digunakan untukmembuat program tersebut adalah bahasa pemrograman. Bahasa pemrograman sangat bermacam-macam: C,C++, Pascal, Java, C#, Basic, Perl, PHP, ASP, JSP, J#, J++ dan masih banyakbahasa lainnya. Dari berbagai bahasa pemrograman cara memberikan instruksinya berbeda-beda namun bertujuan menghasilkan output yang sama.

A. Kriteria Algoritma Menurut Donald E. Knuth

1. Input: algoritma dapat memiliki nol atau lebih inputan dari luar.

2. Output: algoritma harus memiliki minimal satu buah output keluaran.

3. Definiteness(pasti): algoritma memiliki instruksi-instruksi yang jelas dan tidak ambigu.

4. Finiteness (ada batas): algoritma harus memiliki titik berhenti (stopping role).

5. Effectiveness(tepat dan efisien): algoritma sebisa mungkin harus dapat dilaksanakan dan efektif. Contoh instruksi yang tidak efektif adalah: A = A + 0 atau A = A * 1 Namun ada beberapa program yang memang dirancang untuk unterminatable: contoh Sistem Operasi

B. Jenis Proses Algoritma

1. Sequence Process: instruksi dikerjakan secara sekuensial, berurutan.

2. Selection Process: instruksi dikerjakan jika memenuhi kriteria tertentu

3. Iteration Process: instruksi dikerjakan selama memenuhi suatu kondisi tertentu.

4. Concurrent Process: beberapa instruksi dikerjakan secara bersama.

C. Contoh Algoritma

Algoritma menghitung luas persegi panjang:

1.Masukkan panjang (P)

2.Masukkan lebar (L)

3.L ←P * L

4.Tulis L

Hubungan Algoritma dan Struktur Data

Program adalah kumpulan instruksi komputer, sedangkan metode dan tahapan sistematis Dalam program adalah algoritma. Program ini ditulis dengan menggunakan bahasa pemrograman. Jadi bisa kita sebut bahwa program adalah suatu implementasi bahasa pemrograman. Beberapa pakar memberi formula bahwa: program = struktur data + algoritma Bagaimanapun juga struktur data dan algoritma berhubungan sangat erat pada sebuah program. Algoritma yang baik tanpa pemilihan struktur data yang tepat akan membuat program menjadi kurang baik, semikian juga sebaliknya. Menilai Sebuah Algoritma ketika manusia berusaha memecahkan masalah, metode atau teknik yang digunakan untuk memecahkan masalah kemungkinan bisa lebih dari satu. Dan kita memilih mana yang terbaik diantara teknik-teknik itu. Hal ini sama juga dengan algoritma, yang memungkinkan suatu permasalahan dipecahkan dengan metode dan logika yang berlainan.

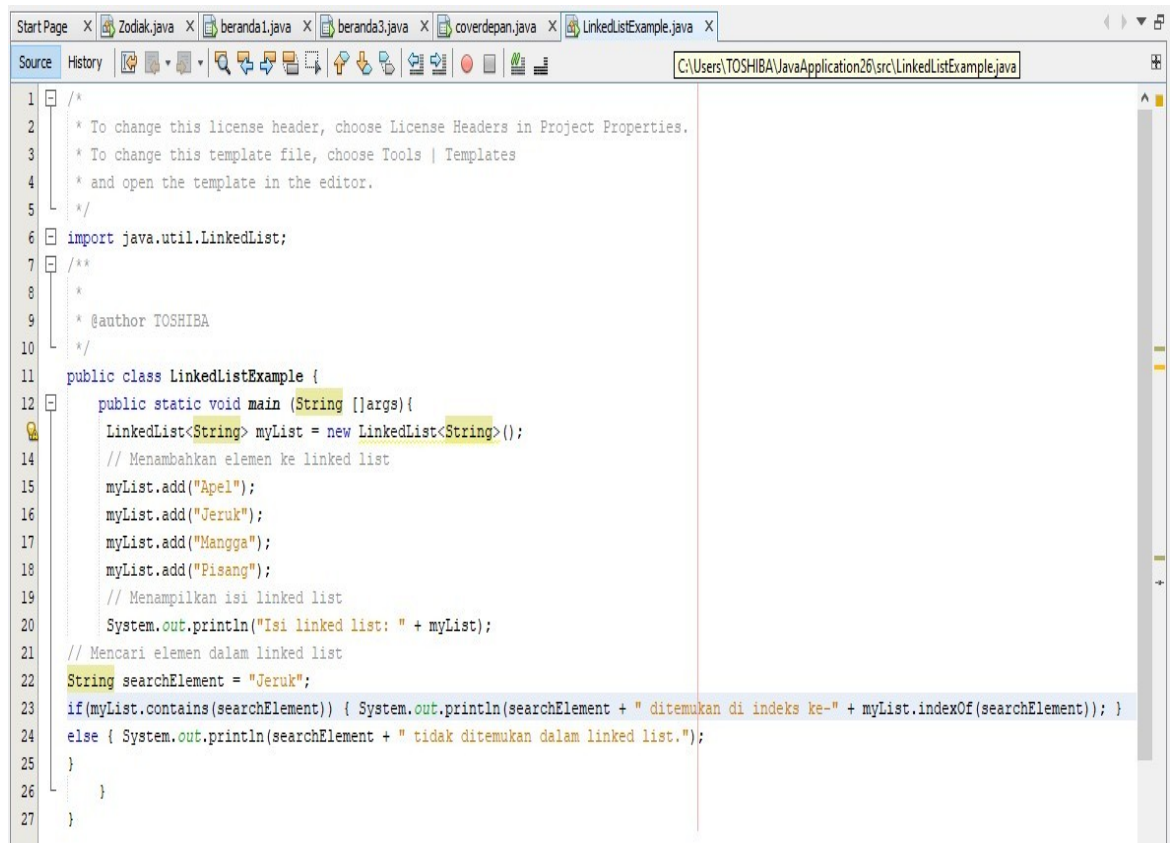
Beberapa persyaratan untuk menjadi algoritma yang baik adalah:

a. Tingkat kepercayaannya tinggi (realibility). Hasil yang diperoleh dari proses harus berakurasi tinggi dan benar.

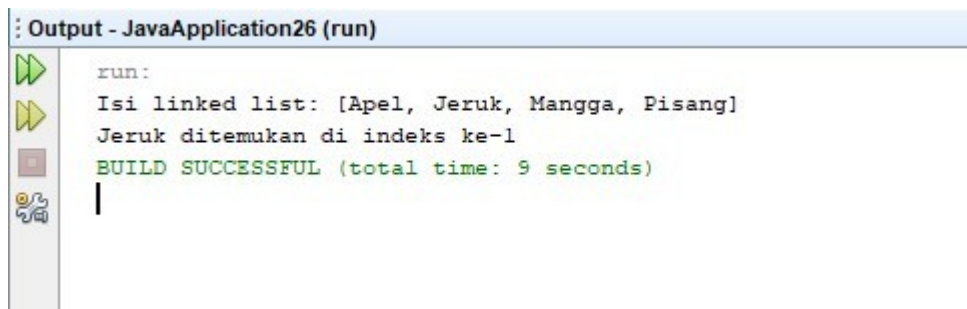
b. Pemrosesan yang efisien (low cost). Proses harus diselesaikan secepat mungkin dan jumlah kalkulasi yang sependek mungkin.

c. Bersifat general. Bukan sesuatu yang hanya untuk menyelesaikan satu kasus saja, tapi juga untuk kasus lain yang lebih general.

d. Bisa dikembangkan (expandable). Haruslah sesuatu yang dapat kita kembangkan lebih jauh berdasarkan perubahan requirement yang ada.

e. Mudah dimengerti. Siapapun yang melihat, dia akan bisa memahami algoritma anda. Sulit dimengertinya suatu program akan membuat sulit pengelolaan.

f. Portabilitas yang tinggi (portability). Bisa dengan mudah diimplementasikan di berbagai platform komputer.

METODE

Struktur data biasanya dapat di gunakan membuat proses pemrograman dengan mudah. dalam segi menyimpan dan mengatur data akan lebih efesien, rapi da terorganisir. Mampu mengelola sumber daya dan layanan, sitem operasinya juga dapat di atur untuk memuat manajemen direktori file, alokasi memori.

Algoritma dapat digunakan untuk membantu memecahkan masalaah tertentu dengan cara yang logis dan sistematis. Algoritma dapat digunakan lebih dari satu kali untuk menyelesaikan masalah yang sama. Algoritma juga dapat digunakan untuk membantu memecahkan masalah yang kompleks dengan lebih sederhana

PEMBAHASAN

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
import java.util.LinkedList;
/**
 *
 * @author TOSHIBA
 */
public class LinkedListExample {
    public static void main (String []args){
     LinkedList<String> myList = new LinkedList<String>();
     // Menambahkan elemen ke linked list
     myList.add("Apel");
     myList.add("Jeruk");
     myList.add("Mangga");
     myList.add("Pisang");
     // Menampilkan isi linked list
     System.out.println("Isi linked list: " + myList);
// Mencari elemen dalam linked list
String searchElement = "Jeruk";
if(myList.contains(searchElement)) { System.out.println(searchElement + " ditemukan di indeks ke-" + myList.indexOf(searchElement)); }
else { System.out.println(searchElement + " tidak ditemukan dalam linked list.");
}
    }
}
```

Gambar 1. contoh program struktur data dan algorima berorientasi objek

Output - JavaApplication26 (run)

```
run:
Isi linked list: [Apel, Jeruk, Mangga, Pisang]
Jeruk ditemukan di indeks ke-1
BUILD SUCCESSFUL (total time: 9 seconds)
```

Gambar 2.output program struktur data dan algoritma berorientasi objek

Program menggunakan linked list , baris 6 -11 : importlibrary linkedlist dan embuat kelas linkedlistexample , baris 12 -13: membuat objek mylist dari kelas linkedlist, baris 15-18 : menambahkan empat elemen ke mylist , baris 20 : menampilkan isi mylist , baris 22-24 : mencari elemen “ jeruk”dalam mylist menggunakan metode contains dan indexof , jika elemen ditemukan indeksnya akan ditampilkan jika tidak ditemukan pesan akan ditampilkan

KESIMPULAN

Struktur Data adalah cara mengumpulkan dan mengatur data sedemikian rupa sehingga kita dapat melakukan operasi pada sebuah data dengan cara yang efektif.

Tipe data struktur data ada beberapa macam yaitu: Grap, tree, array, linked list, dll.
Asal kata Algoritma berasal dari nama Abu Ja’far Mohammed Ibn Musa al-Khowarizmi, ilmuan Persia yang menulis kitab al jabr w’al-muqabala (rules of restoration and reduction) sekitar tahun 825 M.
Algoritma adalah barisan langkah-langkah perhitungan dasar yang mengubah masukan (dari beberapa fungsi matematika) menjadi keluaran.
Bahasa pemrograman dari algoritma sangat bermacam-macam: C,C++, Pascal, Java, C#, Basic, Perl, PHP, ASP, JSP, J#, J++ dan masih banyakbahasa lainnya.

DAFTAR PUSTAKA

NITA AZHAR , 2023, *menguasai dasar-dasar struktur algoritma,* Jakarta selatan: IDS Digital College.

BAMBANG HARIYANTO, 2013, *struktur data memuat dasar pengembangan orientasi objek,* Jakarta Pusat.

TRIVUSI, 2023, *struktur data: pengertian,karatristik, dan kegunaaannya.*

Georgius Rinaldo , 2014, *Algoritma dan struktur data – object pada OOP,* Jakarta.

Greatnusa, 2023, *algoritma adalah pngertian,Karaktristik, Fungsi, Jenisnya,* Jakarta.